Tejero-de-Pablos *et al. EURASIP Journal on Image and Video Processing* (2016) 2016:20
DOI 10.1186/s13640-016-0120-y

EURASIP Journal on Image
and Video Processing

RESEARCH Open Access

Flexible human action recognition in depth video sequences using masked joint trajectories

Antonio Tejero-de-Pablos[1*], Yuta Nakashima[1], Naokazu Yokoya[1], Francisco-Javier Díaz-Pernas[2] and Mario Martínez-Zarzuela[2]

Abstract
Human action recognition applications are greatly benefited from the use of commodity depth sensors that are capable of skeleton tracking. Some of these applications (e.g., customizable gesture interfaces) require learning of new actions at runtime and may not count with many training instances. This paper presents a human action recognition method designed for flexibility, which allows taking users' feedback to improve recognition performance and to add a new action instance without computationally expensive optimization for training classifiers. Our nearest neighbor-based action classifier adopts dynamic time warping to handle variability in execution rate. In addition, it uses the confidence values associated to each tracked joint position to mask erroneous trajectories for robustness against noise. We evaluate the proposed method with various datasets with different frame rates, actors, and noise. The experimental results demonstrate its adequacy for learning of actions from depth sequences at runtime. We achieve an accuracy comparable to the state-of-the-art techniques on the challenging MSR-Action3D dataset.

Keywords: Flexible human action recognition, Runtime learning, Noisy joint trajectory, Depth video sequences

1 Introduction

Human action recognition (HAR) attracts the attention of many researchers due to its numerous applications, such as video surveillance, human computer interaction, and video analysis [1]. However, providing a machine the ability to recognize human actions from an image sequence is a challenging task due to their large variability in various factors [2]. In [3], three main sources of variability are identified: viewpoint, execution rate/speed, and anthropometry.

The recent commodification of depth sensors provides a way to reduce the variability using depth information [4]. They provide 3D structure of scenes, which facilitates the understanding of human actions under conditions in which 2D approaches may be ineffective (e.g., motion perpendicular to the camera plane). Moreover, depth sensors have opened a door for the development of novel techniques that have been used in many computer vision-related research [5, 6]. A distinguished technique, especially advantageous for HAR, is 3D-articulated skeleton tracking in real-time such as [7], which allows modeling human actions in terms of trajectories of body joints. This method is more reliable than using other visual features that are tied to the user's appearance, such as silhouettes. Various techniques have been proposed using depth sensors [8, 9], and more specifically, human joint models. They use different types of classifiers such as hidden Markov models (HMMs) and support vector machines (SVMs).

Most of these recognition methods rely on an expensive learning process with a large training dataset for generalization performance. However, some applications may not count with a large number of instances to be trained with or may need flexibility in learning and classifying the user's behavior (i.e., learning of new actions during runtime). Examples are configurable gesture interfaces or customized retrieval in action databases, in which a new gesture or action can be added to meet the user's needs.

*Correspondence: antonio.tejero.a04@is.naist.jp
[1] Nara Institute of Science and Technology, 8916-5 Takayama, Ikoma city, Nara, 630-0192, Japan
Full list of author information is available at the end of the article

With these premises in mind, we propose and evaluate a novel approach focused on flexibility. Our method is based on the nearest neighbor (NN) approach [10] and uses the joint trajectories estimated from the depth sequences, referred to as action templates (ATs), as a model for each action class. Our method does not require a computationally expensive learning process; modification of the model can be done by just adding new labeled joint trajectories to the set of ATs. For action classification of an unknown action sequence, our method calculates the distance between that sequence and each AT via dynamic time warping (DTW), which is widely used for analyzing time series data [11]. The joint trajectories estimated from depth maps are generally noisy, which might hinder recognition accuracy. For this reason, we include in our ATs the confidence values of each tracked joint along with their respective position, and modify the DTW algorithm to calculate a distance between actions while avoiding erroneous trajectory sections.

The contributions of this work are summarized as follows:

- We propose a novel method for flexible HAR that allows updating the action classifiers at runtime and classification with few training instances. It is designed for applications such as customizable gesture interfaces.
- We also propose a modification of the classification algorithm to mask noisy joint trajectories by using the confidence values from the skeleton tracker.
- We evaluate experimentally the performance of our method and its adequacy for runtime learning of actions in depth sequences. The results demonstrate the effectiveness and accuracy of our method along with its flexibility.

The remainder of this paper is organized as follows. Section 2 introduces the different state-of-the-art techniques in HAR. Section 3 describes our method. Section 4 details the experimental results, and finally, Section 5 concludes this work.

2 Related work

2.1 HAR in RGB video

Until recently, HAR has been performed exclusively on videos captured with traditional cameras [12]. Some methods directly use captured images as spatio-temporal volumes to represent motion [13–15]. In [16], Calderara et al. extracted 1D trajectories from 2D images to represent how an object motion varies in time. However, finer action recognition requires to segment the human body and extract the pose information. Once the body model is obtained from the video, different features related to the human pose can be extracted. Fujiyoshi et al. [17] and Chen et al. [18] extract a primitive skeleton for modeling human actions, in which the skeleton is simplified for reducing the computational cost.

However, these methods suffer from some inaccuracies in the processing of RGB images. According to [19], estimating human poses from 2D video is harsh due to large variations in appearance. In addition, the segmentation of human figures in order to estimate the pose in RGB images is very computationally expensive, due to the high dimensionality of visual features [20]. In the same manner, since the estimation of explicit positions of body parts in a continuous way is difficult, it is also hard to create a general algorithm to learn the model parameters of human actions. It should be noted that the main limitation of such 2D methods is that poses are captured from a single point of view [21], and therefore, certain types of actions can be highly ambiguous.

2.2 HAR in depth video

With the release of commodity depth sensors, HAR underwent a breakthrough thanks to the application of additional 3D information [22]. The use of depth maps alleviates variations in human appearance to a great extent [23]; they can make human segmentation in video far easier and almost immune to illumination, camera blurring, and other factors that hinder HAR. Based on these premises, Li et al. [24] used a depth sensor to obtain a depth map sequence, which is represented as a bag-of-3D points in order to model the actions. Although it outperformed 2D methods, including other bag-of-words-based representations such as [25], this method is still view-dependent because the sampling is performed directly on the depth maps. Another technique involves applying histograms to the 3D point cloud sequences captured by the sensor to calculate descriptors that characterize human shape motion, such as histograms of 4D normals [26] and principal components [27].

One of the advantages of using depth sensors for HAR is that it facilitates the estimation of accurate 3D body joint positions from depth maps via skeleton tracking. These 3D positions can be more direct cues for HAR, providing robustness against variations in viewpoints. Such 3D body joint trajectories used to be available only with expensive equipments such as motion capture devices (MoCap) [28], as in [29]. But currently, they are obtainable with commodity RGB-D sensors with built-in real-time 3D human tracking capabilities (e.g., Microsoft Kinect), although the tracking is not exempt from errors [7]. For example, Xia et al. [8] proposed to use the body joints provided by Kinect to perform HAR using HMMs.

Martínez-Zarzuela et al. [30] and Wang et al. [9] use the discrete Fourier transform to represent the joint trajectories in the frequency domain and then feed them into a classifier, Fuzzy ARTMAP [31] and support vector machines (SVMs) [32], respectively. The discrete Fourier transform reduces the dimensionality of the joint trajectories by assuming that the most crucial information is concentrated in the lower frequency components. It also reduces noises due to tracking errors, which is a problem inherent to joint estimation from depth maps.

Variations in execution rate of human actions have a negative impact in HAR [33]. Many works have relied on DTW to gain robustness against these variations. Müller and Röder [34] used DTW to build semantically interpretable action models by extracting relational features that encode temporal dynamics. These relational features (e.g.„ the right hand is up or down) exclude a lot of detail of the action, but retain view-invariant information about the overall configuration of a pose for its classification. However, because of the loss of detail, this method confuses actions when they are too similar or too short, and the accuracy is very dependent on the manually designed features. In [35], Wang and Wu dealt with variations in execution rate by combining an SVM-based classification algorithm with DTW. Alternatively to DTW, the longest common subsequence (LCSS) is used in [36] to make their action classifier invariant to temporal variations. In [37], the authors find a representation of the body joint trajectories that is robust against execution rate variations among subjects. They consider two HAR schemes, a NN classifier, and an SVM classifier.

2.3 Flexible HAR applications

There is a range of HAR-based applications that require learning new actions in runtime. Applications such as customizable gesture interfaces [38, 39] and action databases, either for indexing or retrieval [34, 40], can benefit from such capability, since they are expected to be able to recognize a new type of action right after being input. This kind of applications also does not count with many learning instances [41]. Hence in this work, we consider the flexibility of approach by two factors:

- Being able to learn a certain action class at runtime.
- Being able to recognize actions even with a very small number of training instances.

We consider that a method is capable of runtime learning if it does not perform any optimization of the classifier when learning a new action instance. The majority of the previously mentioned works rely on classifiers with a costly learning process that cannot be updated at runtime (e.g., SVM) and therefore are not suitable for applications that require adaptive modification of the training model. On the other hand, methods that are capable of runtime learning (e.g., NN) allow this, but to the best of our knowledge they have not proved state-of-the-art accuracy yet. We propose a HAR method that, while facilitating joint trajectories obtained from depth map sequences, focuses on achieving flexible recognition with an accuracy comparable to the state-of-the-art methods. In addition, in order to gain robustness against noise, we use the joint estimation confidence value during classification.

3 Flexible HAR using masked joint trajectories

Figure 1 depicts an overview of our method, which takes a nearest neighbor-based approach to gain flexibility instead of learning a classifier for each action class. We first estimate the 3D joint positions using skeleton tracking from a series of depth map sequences using, e.g., [42], and store them with their action labels as instances of a training dataset. One of the main issues that lead to failure in HAR is concerned with the estimation errors in the skeleton tracking, as stated in [9]. Fortunately, the joint position estimation algorithm provides a confidence value for each joint tracked in each frame. Our method uses it for both learning and recognition stages to alleviate the problem of erroneous skeleton tracking. Then, we prepare an AT for each given action class, which can be viewed as a model of a specific action. Each AT consists of a set of joint trajectories of the action instances belonging to that class along with the confidence values for each joint positions.

At the recognition stage our method tracks the joint trajectories of an unknown action instance in the same way as the learning process, and retrieves its closest instance from the ATs in the database. Since different instances of the same action can be subjected to temporal variations (especially different length and execution speed), we employ a DTW-based distance measure for template matching during the nearest neighbor-based classification.

3.1 Action template learning

To generate an AT, we manually select $J = 15$ different joints from the skeleton tracked in an action instance, as illustrated in Fig. 2. Let $\mathbf{p}'_{fj} = (x_{fj}, y_{fj}, z_{fj})^\top$ denote the 3D position of joint j at frame f. Since these positions are in the RGB-D sensor's coordinate system, they can vary from one action instance to another depending on the position of the actor relative to the sensor. For reducing this variability, we transform the joint coordinates so that a certain joint coincides with the origin to improve the robustness against viewpoint variations. In this work, we choose the torso as the origin, thus denoting the transformed joint position as $\mathbf{p}_{fj} = \mathbf{p}'_{fj} - \mathbf{p}'_{f\text{torso}}$.

The joint trajectories of all the instances from a certain action class are then aggregated to form an AT. Along

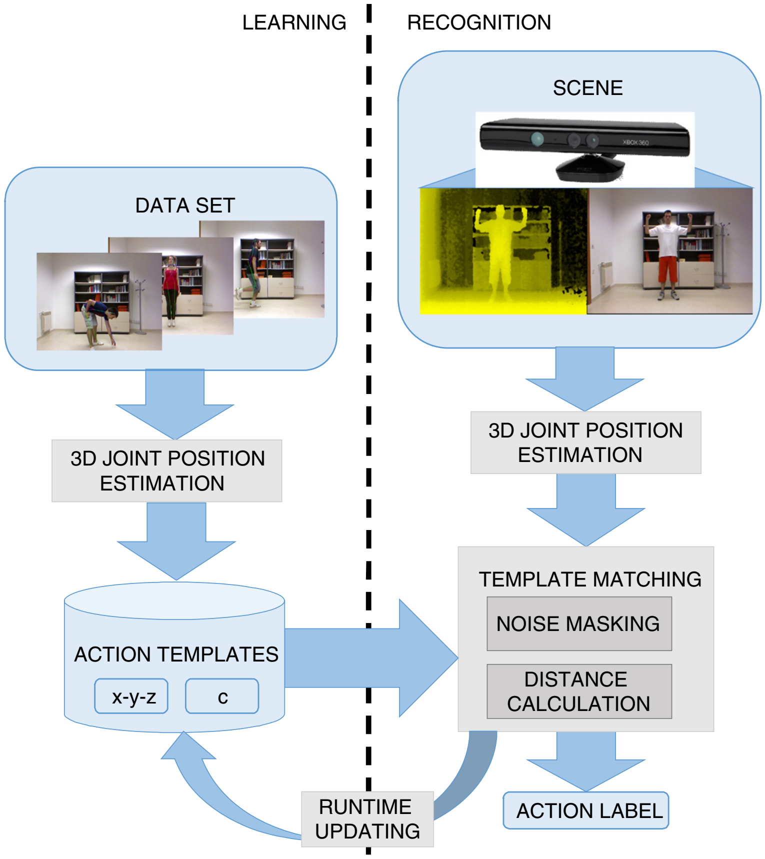
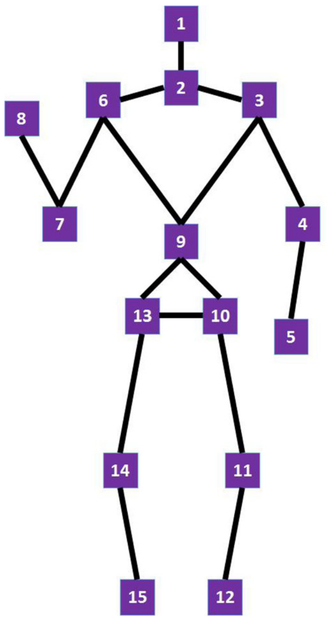


Fig. 1 Overview of our HAR method. The 3D joint positions (*x*,*y*,*z*) along with the confidence value (*c*) are tracked from the video source to build action templates for each action class. They are used to match new actions and updated at runtime



1	Head
2	Neck
3	Left shoulder
4	Left elbow
5	Left hand
6	Right shoulder
7	Right elbow
8	Right hand
9	Torso
10	Left hip
11	Left knee
12	Left foot
13	Right hip
14	Right knee
15	Right foot

Fig. 2 Human body joints used in our method. Numbering and names correspond to the ones used in the skeleton tracker

with them, the associated confidence values of the tracked positions offered by the joint estimation algorithm of the skeleton tracker [42] are also included. Let m_i be the action class label for the joint trajectories of the instance i in the training dataset (m_i = *running*, for example), $P_i = \{\mathbf{p}^i_{fj} | f = 1, \ldots, F_i, j = 1, \ldots, J\}$ the corresponding joint trajectories, and $C_i = \{\mathbf{c}^i_{fj} | f = 1, \ldots, F_i, j = 1, \ldots, J\}$ their corresponding confidences, where F_i is the number of frames for action instance i. The AT for action class M is then a set of joint trajectories with their respective confidence values, i.e.,

$$A_M = \{(P_i, C_i) | is.t. m_i = M\}. \quad (1)$$

The learning process only requires the generation of ATs.

3.2 Action classification

Our recognition process calculates a distance measure to find in our ATs the action instance that is the nearest neighbor of the given unknown instance. Due to the variability in the execution of human actions, naive distance measures are not applicable. For this reason, we employ the use of a DTW-based distance measure, which does not require temporal alignment nor synchronization between a pair of sequences in different sizes [11].

Let $U = \{\mathbf{u}_{fj} | f = 1, \ldots, F_{\mathrm{U}}, j = 1, \ldots, J\}$ be the joint trajectories of an unknown action instance, with F and J as the total number of frames of the action and the number of joints, respectively. Note that length F_{U} of an unknown action and length F_i of an action instance in an AT are generally different. The local distance between the positions of joint j in frame f of U and frame f' in P_i is defined as the Euclidean distance as follows:

$$e(\mathbf{u}_{fj}, \mathbf{p}^i_{f'j}) = \|\mathbf{u}_{fj} - \mathbf{p}^i_{f'j}\|_2. \tag{2}$$

Then, using confidence value c_{fj} generated during the tracking, we apply a mask to the trajectory of each joint j for each frame f. If this value is smaller than a predefined threshold τ, we determine that that part of the trajectory is not useful for classification. Therefore, we assign a binary weight to each point of a joint trajectory by

$$w_{fj} = \begin{cases} 1 & \text{if } c_{fj} \geq \tau \\ 0 & \text{otherwise} \end{cases}. \tag{3}$$

This weighting is applied to the joint positions of both U and P_i. This means only J' out of the J joints are used for frame f, where J' is the number of joints that are not masked ($J' \leq J$). Thus, we define the masked distance between all joint positions $\mathbf{u}_f$ and $\mathbf{p}^i_{f'}$ in frames f and f' as

$$d(\mathbf{u}_f, \mathbf{p}^i_{f'}) = \frac{1}{J'} \sum_{j=1}^{J} e(\mathbf{u}_{fj}, \mathbf{p}^i_{f'j}) w_{fj} w_{f'j}. \tag{4}$$

Using this distance, the DTW-based distance measure between U and P_i is defined as the minimum sum of the local distances over a warping path. Namely, letting $\mathbf{t}_n = (f_n, f'_n)$ be a pair of frames, f for the unknown action instance U and f' for the one in an AT, and $T = \{\mathbf{t}_n | n = 1, \ldots, N\}$ a warping path over which the sum is calculated, the DTW-based distance D is given by

$$D(U, P_i) = \min_T \sum_{(f_n, f'_n) \in T} d(\mathbf{u}_f, \mathbf{p}^i_{f'}) \tag{5}$$

$$\begin{aligned} \text{subject to } \; & \mathbf{t}_1 = (1,1) \;\text{ and }\; \mathbf{t}_N = (F_{\mathrm{U}}, F_i) \\ & f_1 = 1 \leq f_2 \leq \cdots \leq f_N = F_{\mathrm{U}} \\ & f'_1 = 1 \leq f'_2 \leq \cdots \leq f'_N = F_i \\ & \mathbf{t}_{n+1} - t_n \in \{(1,0), (0,1), (1,1)\}. \end{aligned} \tag{6}$$

Equation (5) can be minimized by dynamic programming.

Since the nearest neighbor-based approach needs to compare the distances calculated for action instances of different length, a normalized version of this distance is calculated. The normalizing factor in this case is the length of the warping path T, that is

$$D'(U, P_i) = \frac{1}{N} D(U, P_i). \tag{7}$$

The action class m^* for the unknown action instance U is given as the one whose AT includes an action instance that gives the minimum distance with U, i.e.,

$$m^* = m_{i^*} \text{ where } i^* = \arg\min_i D'(U, P_i). \tag{8}$$

4 Experimental results

In order to evaluate our approach for generic HAR, we choose datasets containing heterogeneous actions [43] involving the whole body. More specifically, we used the CMU MoCap dataset, the MSR-Action3D dataset and our self-generated dataset, and compared the results with other state-of-the-art methods. A sample frame of each one is shown in Fig. 3. No ethical approval was needed for these experiments, since the individual recording time of the UGOKI3D dataset was short and there was not risk of any damage or embarrassment. The person in Fig. 3 also approved appearing in the figure. Besides, all participants of the UGOKI3D dataset gave their consent to using the captured data in these experiments.

4.1 Implementation details

The recognition algorithm was implemented in Matlab, running in Windows 8 (64 bit), installed in a PC with an Intel Core i7 processor and 16 GB RAM. In addition, for the experiments, we used an empirically determined threshold value $\tau = 0.1$.

Fig. 3 Example image of the datasets used. *Left*: self-generated, *center*: CMU MoCap, *right*: MSR-Action3D

4.2 Self-generated UGOKI3D dataset

The UGOKI3D dataset was generated using a Microsoft Kinect v1 for evaluating our previous HAR method, which used the discrete Fourier transform and neural networks [30]. It is comprised of eight heterogeneous actions that involve all body parts, and with different characteristics: periodic, aperiodic, static (the location of the user in the scene does not vary) and non-static. The actions are performed by nine actors of different gender and appearances: (a) *bending*, (b) *jumping jacks*, (c) *jumping-forward*, (d) *jumping*, (e) *side-galloping*, (f) *walking*, (g) *waving one hand*, and (h) *waving both hands*. For the sake of comparability, we used the same evaluation scheme, applying leave-one-out (LOO) cross validation, in which we trained our model with sequences of eight actors and evaluated our proposed method with the sequences of the remaining one actor. The accuracy was averaged over all nine iterations.

The average accuracy rate obtained in this experiment was 94.44 %, which is higher than the one achieved with our previous method (93.05 %). The confusion matrix for all actions is shown in Table 1, whose rows and columns indicate the ground truth and recognition results, respectively. As it can be observed, the most common classification errors involved actions that present similar fast position variations in the lower body, i.e., *jumping-forward* and *walking*. One of the reasons of these inaccuracies is the occasional errors in the skeleton tracking.

4.3 CMU MoCap dataset

To show the potential performance of our proposed method when the skeleton tracking is almost perfect, we used the motion capture dataset provided by Carnegie Mellon University, which contains actions captured at 120 fps [28]. This dataset was not generated from sequences captured with depth sensors, but with a motion capture technique using markers attached to the human body. This dataset is composed by multiple actors performing heterogeneous actions divided in categories such as locomotion and sports. However, not all the actors perform every action, and the number of instances of each action can vary largely. To be consistent with the experiment in the previous section, a subset of eight different actions was selected, with a noticeable emphasis on the lower body, i.e. (a) *running*, (b) *walking*, (c) *jumping-forward*, (d) *jumping*, (e) *soccer kick*, (f) *boxing*, (g) *jumping jacks*, and (h) *hand signs*. Also, although the dataset offers joint trajectories in more than 20 body parts, we use its subset that corresponds to the 15 joints of our UGOKI3D dataset. In addition, since this skeleton tracking method does not provide a confidence parameter, we did not use masking for this experiment ($w_{fj} = 1$).

Our method was evaluated applying LOO cross validation again, achieving the accuracy of 97.22 %. The accuracy for each action is summarized in the confusion matrix of Table 2. Only the *jumping jacks* action is misclassified twice; in one sequence, the actor only performed half a repetition, and in the other the actor did not move the arms accordingly to the action. As expected, due to the accurate joint estimates, the results of this experiments were highly accurate, regardless of the types of actions. We also evaluated our previous method [30], resulting in an inferior accuracy of 91.67 %.

4.4 MSR-Action3D dataset

The MSR-Action3D dataset includes various challenging actions and has been widely used to evaluate HAR methods. This dataset contains twenty different static

Table 1 Confusion matrix for the UGOKI3D dataset

	(a)	(b)	(c)	(d)	(e)	(f)	(g)	(h)
(a)	100 % (9/9)							
(b)		88.89 % (8/9)					11.11 % (1/9)	
(c)			88.89 % (8/9)		11.11 % (1/9)			
(d)				100 % (9/9)				
(e)					100 % (9/9)			
(f)			11.11 % (1/9)		11.11 % (1/9)	77.78 % (7/9)		
(g)							100 % (9/9)	
(h)								100 % (9/9)

Table 2 Confusion matrix for the CMU MoCap dataset

	(a)	(b)	(c)	(d)	(e)	(f)	(g)	(h)
(a)	100 % (9/9)							
(b)		100 % (9/9)						
(c)			100 % (9/9)					
(d)				100 % (9/9)				
(e)					100 % (9/9)			
(f)						100 % (9/9)		
(g)				11.11 % (1/9)			77.78 % (7/9)	11.11 % (1/9)
(h)								100 % (9/9)

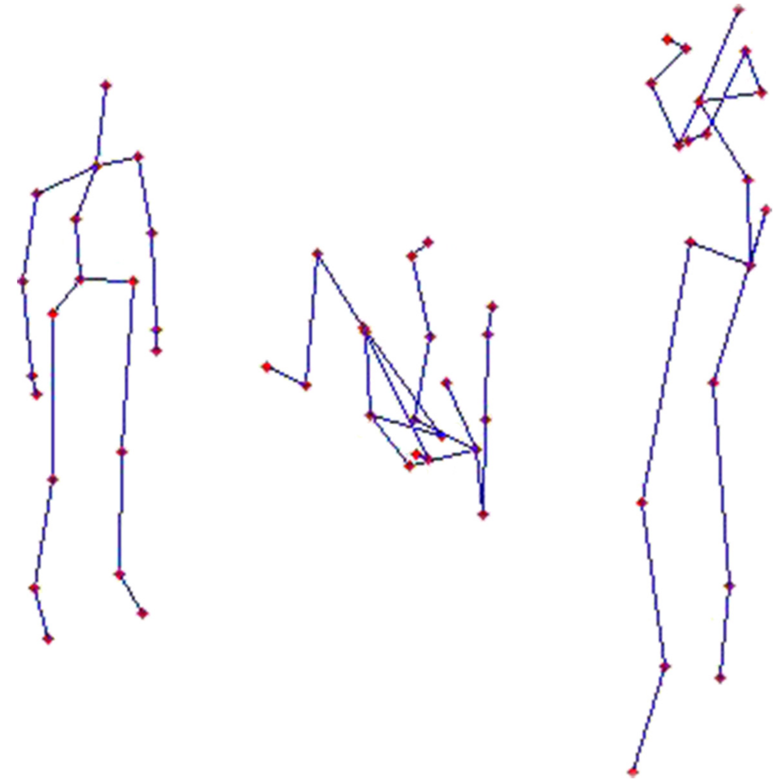


Fig. 4 Noise present in the MSR-3DAction dataset. *Left*: base position, *center* and *right*: noisy samples

actions performed by up to ten actors, and the same actor did the same action from one to three times. The actions are (a) *high arm wave*, (b) *horizontal arm wave*, (c) *hammer*, (d) *hand catch*, (e) *forward punch*, (f) *high throw*, (g) *draw x*, (h) *draw tick*, (i) *draw circle*, (j) *hand clap*, (k) *two hand wave*, (l) *side-boxing*, (m) *bend*, (n) *forward kick*, (o) *side kick*, (p) *jogging*, (q) *tennis swing*, (r) *tennis serve*, (s) *golf swing*, and (t) *pickup and throw*. The dataset was built using sequences captured with depth sensors at 15 fps. It provides the 3D position and the tracking confidence of 20 joints per frame, but we kept using 15 joints for our proposed method since we considered the extra five (wrists, ankles, and center hip) do not add much information to the model. Although some works highlight that its difficulty resides in the similarity of its actions, in our opinion, the dataset is challenging due to the noise present in the skeleton tracking. Figure 4 shows some unrealistic poses included in the dataset.

We followed the evaluation methodology employed in previous works [9, 24, 26, 37], and divided the 555 instances into three groups as shown in Table 3. For each group, we conducted a cross-subject experiment in which the actions performed by actors 1, 3, 5, 7, and 9 were used for training and the ones from actors 2, 4, 6, 8, and 10 for testing. The overall recognition accuracy obtained in the experiment was 84.09 %. The individual accuracy rates for SS1, SS2, and SS3 are 80 %, 78.57 %, and 93.69 % , respectively. The first two subgroups were more erroneous than the third one. These results are shown in detail in Tables 4, 5, and 6.

Table 7, obtained partially from [9], shows the generalization performance of our method compared with other state-of-the-art methods that were evaluated against this dataset using the same configuration. The upper part of the table lists the methods that are capable of runtime learning (e.g., NN), and the lower part of the table lists the ones that are not (e.g., SVM). Our method's accuracy outperforms the other HAR methods that are

Table 3 Action subdivision of the MSR-Action3D dataset used in the experiments

Subset 1 (SS1)	Subset 2 (SS2)	Subset 3 (SS3)
Horizontal arm wave	High arm wave	High throw
Hammer	Hand catch	Forward kick
Forward punch	Draw x	Side kick
High throw	Draw tick	Jogging
Hand clap	Draw circle	Tennis swing
Bend	Two hand wave	Tennis serve
Tennis serve	Forward kick	Golf swing
Pickup and throw	Side boxing	Pickup and throw

Table 4 Confusion matrix for the MSR-Action3D dataset (SS1)

	(b)	(c)	(e)	(f)	(j)	(m)	(r)	(t)
(b)	50 % (6/12)	8.33 % (1/12)	41.67 % (5/12)					
(c)		75 % (9/12)	25 % (3/12)					
(e)			100 % (11/11)					
(f)	18.18 % (2/11)	9.09 % (1/11)		72.73 % (8/11)				
(j)					100 % (15/15)			
(m)						46.67 % (7/15)		53.33 % (8/15)
(r)							100 % (15/15)	
(t)							7.14 % (1/14)	92.86 % (13/14)

Table 6 Confusion matrix for MSR-Action3D dataset (SS3)

	(f)	(n)	(o)	(p)	(q)	(r)	(s)	(t)
(f)	81.82 % (9/11)			18.18 % (2/11)				
(n)		100 % (15/15)						
(o)			90.91 % (10/11)	9.09 % (1/11)				
(p)				100 % (15/15)				
(q)					100 % (15/15)			
(r)						100 % (15/15)		
(s)							100 % (15/15)	
(t)					28.57 % (4/14)			71.43 % (10/14)

capable of runtime learning by far, and is very close to the state-of-the-art methods. Compared with the other two datasets used, the MSR-Action3D has a larger presence of tracking noise. As Müller and Röder remarked in [34], when performing HAR with noisy templates, recognizing new actions becomes hard (see Table 5). However, when we apply the confidence value of the skeleton tracker to avoid using the erroneous sections in the AT, matching the recognition performance of our method improves noticeably, as shown in Table 7.

4.5 Flexible HAR

We evaluate the performance of our proposed method's capability of learning new action instances in runtime. We assume a scenario of a customizable gesture interface for a certain application system, in which a command for the system is issued via the gesture interface whose backend is our HAR method. This scenario supposes that the gesture interface has a predefined set of gestures, each of which has a single instance of the corresponding gesture when initialized. The interface learns at runtime; if the interface fails in correctly recognizing an input instance of a gesture, the user specifies the correct label of the instance and the interface includes it to the corresponding AT.

To demonstrate the performance under this scenario, we used the action classes contained in each subset of the MSR-Action3D dataset instead of actual gestures (eight different action classes per subset). We used 20 action

Table 5 Confusion matrix for the MSR-Action3D dataset (SS2)

	(a)	(d)	(g)	(h)	(i)	(k)	(l)	(n)
(a)	83.33 % (10/12)	8.33 % (1/12)	8.33 % (1/12)					
(d)	50 % (6/12)	16.67 % (2/12)	16.67 % (2/12)				16.67 % (2/12)	
(g)			92.31 % (12/13)	7.69 % (1/13)				
(h)	20 % (3/15)			80 % (12/15)				
(i)	26.67 % (4/15)		13.33 % (2/15)		60 % (9/15)			
(k)						100 % (15/15)		
(l)		6.66 % (1/15)					86.68 % (13/15)	6.66 % (1/15)
(n)								100 % (15/15)

Table 7 Recognition accuracy comparison for the MSR-Action3D dataset

Method	Accuracy	Type
Proposed method	84.09 %	Skeleton
Proposed method (no noise masking)	79.31 %	Skeleton
Rate-invariant analysis (NN) [37]	63 %	Skeleton
Dynamic temporal warping [34]	54 %	Skeleton
MMTW [35]	92.57 %	Skeleton
Joint movement similarities [36]	91.2 %	Skeleton
HOPC [27]	90.9 %	Depth
Rate-invariant analysis (SVM) [37]	89 %	Skeleton
HON4D [26]	88.36 %	Depth
Mining actionlet ensemble [9]	88.2 %	Skeleton
Histograms of 3D joints [8]	78.97 %	Skeleton
Action graph on bag-of-3D points [24]	74.7 %	Depth
Hidden Markov model [29]	63 %	Skeleton
Recurrent neural network [46]	42.5 %	Skeleton

instances of each action class in the subset, and divided it into two groups: ten for learning and ten for testing. That is, for each subset we use a learning and testing groups of 80 action instances each. At the start, we generate the ATs with a single instance for each class, and then we feed the remaining instances in the learning group one by one (72 instances in total). If our HAR method fails to recognize one instance, it adds that instance to the corresponding AT. We evaluated the accuracy of the method using the test set after an instance in the learning group is input. We repeat this 100 times, randomizing the instances in the learning and testing groups, and the order of the input learning instances. The recognition accuracy is the average of all repetitions. We also measured the time required for recognizing the instances in the test set, which is also averaged over the 100 repetitions.

Figure 5 shows the runtime accuracy of our method for each instance in the learning group evaluated against the test group. The final recognition accuracies achieved for subsets SS1, SS2, and SS3 are 75.12, 79.06, and 88 %, respectively, with 37, 35, and 27 instances on average added to the ATs, respectively (see Fig. 6). By comparing these results to the previous experiment, it can be noticed that our method is able to provide a similar accuracy generating ATs in runtime with less than half the action instances than the previous configuration. It is also remarkable the fact that our method achieves accuracies around 50 % with just a single instance per action class. Figure 7 shows the time in seconds spent in classifying one gesture using our implementation. It grows from 0.5 to about 2 s almost linearly as the number of learned instances in our ATs grows.

4.6 Discussion

Our experimental results have shown that our approach can be successfully applied for HAR at runtime in depth video sequences. In comparison with many related works, we use raw 3D joint trajectories instead of other representations [9, 30, 34] such as Fourier transform, joint mining, or boolean features, thereby reducing the computational cost of learning. By applying DTW, we gain robustness against variations in execution rates, which heavily affect HAR. Although this methodology is more sensitive to the noise present in the joint position estimation, we manage to effectively alleviate this problem by using the confidence values provided by the skeleton tracker itself. We achieved high recognition rates for a wide variety of actions (periodic, static, etc.) and sensors (high frame rate, low frame rate), and outperformed other methods that are capable of runtime learning on the challenging MSR-Action3D dataset.

Compared to the state-of-the-art methods that are not capable of runtime learning, our performance is slightly inferior. We consider the reason is that we do not rely on an intricate training phase in order to reduce the cost of learning a new action instance. Besides, our feature set consists of a small number of joint positions tracked in real-time, with no other RGB/depth information. Basically, our method deals with a trade-off between flexibility

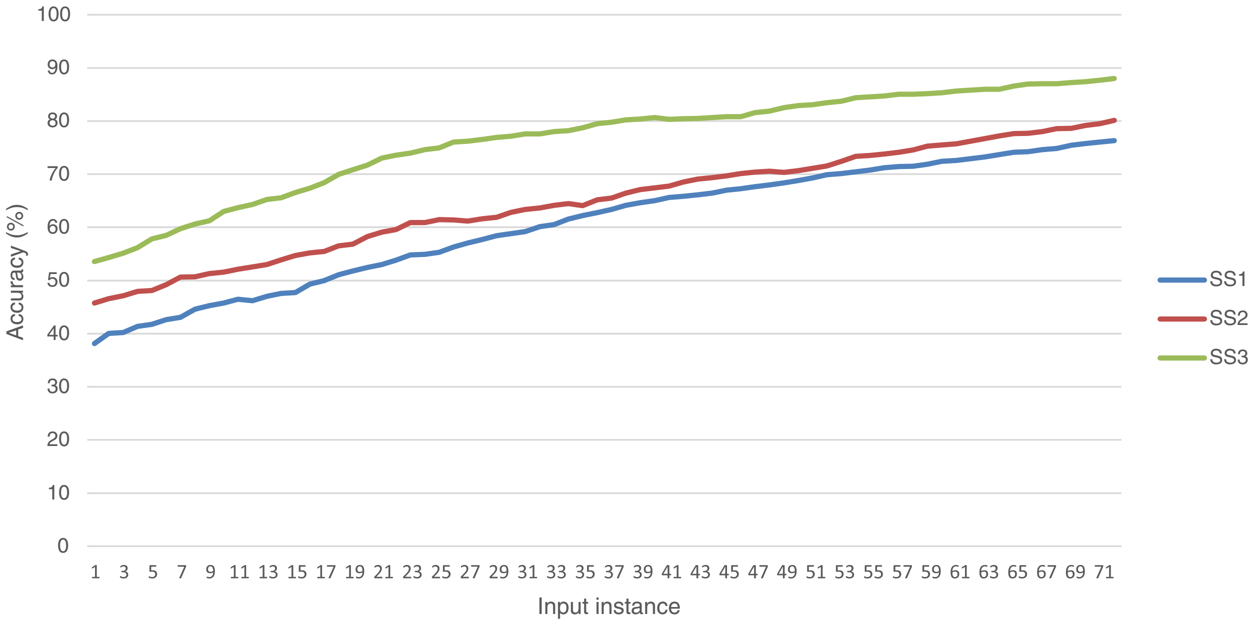


Fig. 5 Recognition accuracy during runtime learning. *Horizontal axis*: input instances, *vertical axis*: average recognition accuracy (percentage)

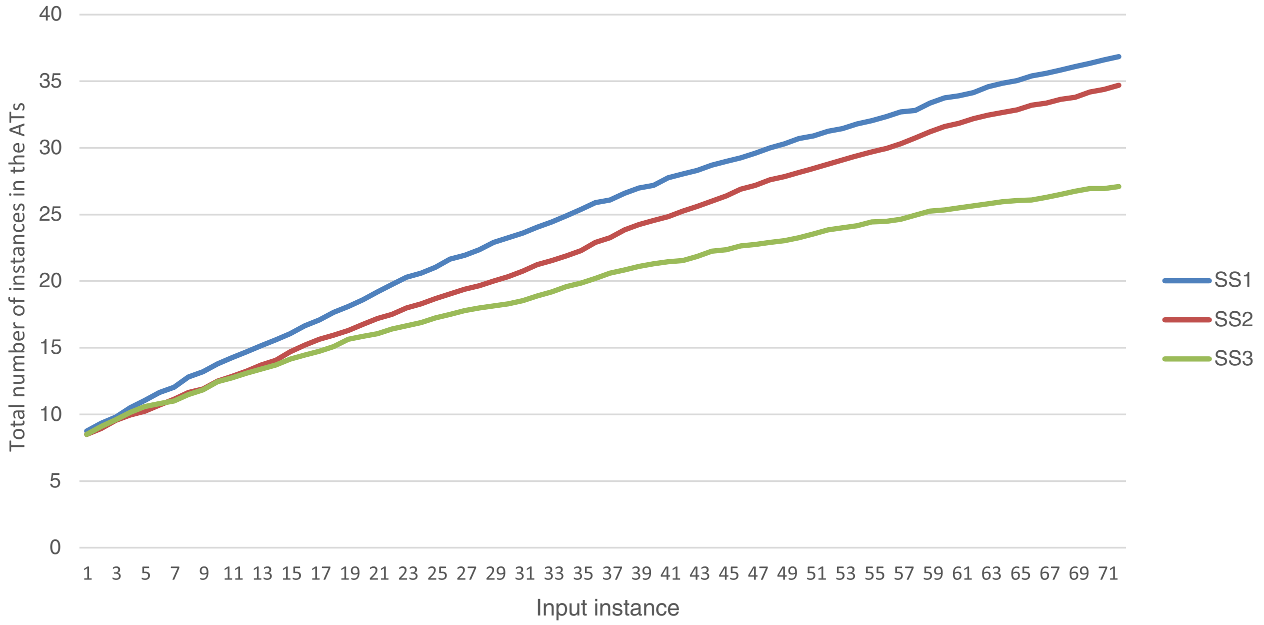


Fig. 6 Total number of instances in the ATs during runtime learning. *Horizontal axis*: input instances, *vertical axis*: average number of instances in the ATs

and accuracy in order to allow for runtime learning. For example, Wang and Wu [35] also deal with variations in execution rate of actions using a human joint model. But contrary to our proposed method, their maximum margin temporal warping (MMTW) method relies on a costly SVM algorithm in order to extract the optimal template for the training dataset. Therefore, it can be considered that MMTW is not suitable for runtime learning of new action instances. Also, in their skeleton approach, they use 1140 (20× 19× 3) features per frame which is the distance in the three-dimensional space of each body joint offered by Kinect to the rest. In order to maintain the computational efficiency, our method uses only 45 (15× 3) features per frame. The same can be said for other works [9, 36, 37].

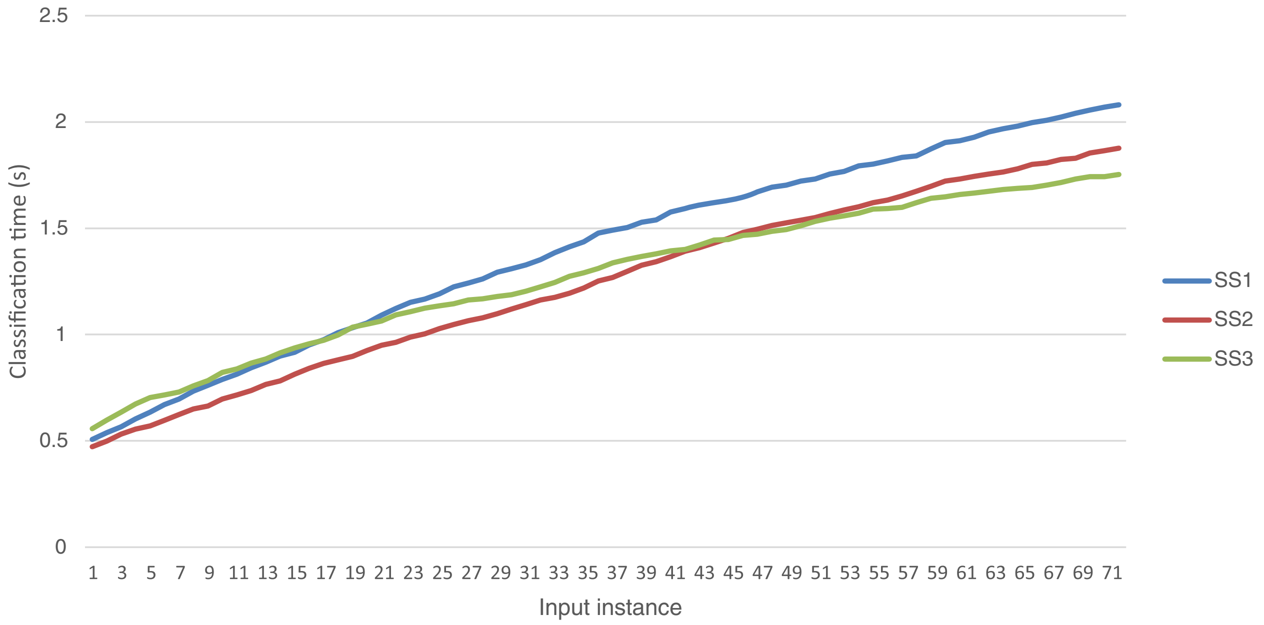


Fig. 7 Classification time for one instance during runtime learning. *Horizontal axis*: input instances, *vertical axis*: average classification time (seconds)

Besides, we have proved experimentally that our method offers a great flexibility that would allow users to provide some feedback on wrong classifications or even to add a new action category at runtime. Also, an AT can contain instances for several ways of performing the same action class (e.g., drinking with your left hand or right hand, gesturing standing up or sitting, etc.), which provides robustness against variations in the way actions are executed. Another example of its flexibility is that, in case of performing action recognition of a specific body part, the number of trajectories used can be easily modified, generating customized ATs with just the joints of interest (hands, legs, etc.). Also, the joint positions contained in an AT itself can be used to reproduce the captured action, which is useful for animation purposes.

This work also shows an effective way for applying DTW to action recognition. To the best of our knowledge, the previous results of using exclusively DTW in a 3D joint-based HAR method have not been convincing enough [44]. Although intuitively DTW fits quite well a task such as analyzing action trajectories, it has been criticized arguing that it is more sensitive to temporal scale changes than HMM-based methods [29], and produces large temporal misalignments in case of periodic actions [9]. But rather than that, by looking at the actions used in the experiments and the results obtained, we inferred that what most affects this technique was the noise in the skeleton tracking process.

When the number of instances in our ATs increases, the time cost of our method in order to classify one action can be high (Fig. 7) due to the computational cost of DTW, $O(MN)$, where M and N are the lengths of the two compared sequences [45]. However, implementing a real-time system would not be infeasible due to the increasing speed of computers and acceleration techniques based on parallel execution such as GPGPU, given the fact that in our algorithm distance calculations can be executed completely in parallel. Our method has also the advantage of not requiring a large number of action instances.

5 Conclusions

In this paper, we have presented a flexible method for recognizing actions from trajectories estimated from depth sequences based on the generation of action templates using joint trajectories. To deal with inaccuracies in the joint position estimation, our method integrates a mask for the noisy sections of the trajectories during classification using the confidence values offered by the 3D joint position estimation algorithm [42]. The proposed method deals with a trade-off between flexibility and accuracy, achieving comparable results with the state-of-the-art methods in a challenging dataset. We have also successfully demonstrated the flexibility of our approach, which allows performing HAR with very few training instances, while learning new actions at runtime. This is a very powerful feature in applications such as action databases, video analysis, and customizable gesture interfaces.

For future work, we plan to optimize the generation of action templates by eliminating redundant information (i.e., clustering similar instances or forgetting unused instances), and therefore reducing classification times. We will also address the recognition of action classes that only differ in their speed (e.g., touching and punching). We aim to evaluate in depth the flexibility of our approach in a specific application such as the aforementioned, using video streams from depth cameras.

6 Consent

All the participants of the self-generated UGOKI3D dataset gave their consent to using this dataset for the experiments published in this paper.

Competing interests

The authors declare that they have no competing interests.

Authors' contributions

ATdP designed the core methodology of the study and carried out the implementation and experiments. YN participated in the design of the human action recognition methodology and helped to draft the manuscript. NY participated in the design of the online learning experiment and helped to draft the manuscript. FJDP designed the experiments with the UGOKI3D dataset and participated in the coordination of the study. MMZ conceived of the study and participated in the creation of the UGOKI3D dataset. All authors read and approved the final manuscript.

Acknowledgements

This work is partly supported by a Microsoft Research CORE Grant and JSPS KAKENHI No. 16K16086.

Author details

[1] Nara Institute of Science and Technology, 8916-5 Takayama, Ikoma city, Nara, 630-0192, Japan. [2] University of Valladolid, Campus Miguel Delibes, Paseo Belén 15, Valladolid, 47011, Spain.

Received: 10 August 2015 Accepted: 20 April 2016
Published online: 03 June 2016

References

1. P Turaga, R Chellappa, VS Subrahmanian, O Udrea, Machine recognition of human activities: a survey. IEEE Trans. Circuits Syst. Video Technol. **18**(11), 1473–1488 (2008)
2. J Aggarwal, MS Ryoo, Human activity analysis: a review. ACM Comput. Surv. **43**(3), 16–11643 (2011)
3. Y Sheikh, M Sheikh, M Shah, in *Proc. the 10th IEEE International Conference on Computer Vision (ICCV)*. Exploring the space of a human action, vol. 1, (2005), pp. 144–149
4. C Plagemann, V Ganapathi, D Koller, S Thrun, in *Proc. the IEEE International Conference on Robotics and Automation (ICRA)*. Real-time identification and localization of body parts from depth images, (2010), pp. 3108–3113
5. J Giles, Inside the race to hack the kinect. New Scientist. **208**(2789), 22–23 (2010)
6. EA Suma, B Lange, A Rizzo, DM Krum, M Bolas, in *Proc. the IEEE Virtual Reality Conference (VR)*. FAAST: The flexible action and articulated skeleton toolkit, (2011), pp. 247–248
7. J Shotton, T Sharp, A Kipman, A Fitzgibbon, M Finocchio, A Blake, M Cook, R Moore, Real-time human pose recognition in parts from single depth images. Commun. ACM. **56**(1), 116–124 (2013)

8. L Xia, C-C Chen, J Aggarwal, in *Proc. the IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*. View invariant human action recognition using histograms of 3d joints, (2012), pp. 20–27
9. J Wang, Z Liu, Y Wu, J Yuan, in *Proc. the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*. Mining actionlet ensemble for action recognition with depth cameras, (2012), pp. 1290–1297
10. S Cost, S Salzberg, A weighted nearest neighbor algorithm for learning with symbolic features. Mach. Learn. **10**(1), 57–78 (1993)
11. L Rabiner, B-H Juang, Fundamentals of speech recognition (1993)
12. PVK Borges, N Conci, A Cavallaro, Video-based human behavior understanding: a survey. IEEE Trans. Circuits Syst. Video Technol. **23**(11), 1993–2008 (2013)
13. E Shechtman, M Irani, in *Proc. the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*. Space-time behavior based correlation, vol. 1, (2005), pp. 405–412
14. D-Y Chen, S-W Shih, H-Y Liao, in *Proc. the IEEE International Conference on Multimedia and Expo*. Human action recognition using 2-D spatio-temporal templates, (2007), pp. 667–670
15. H Meng, N Pears, C Bailey, in *Proc. the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*. A human action recognition system for embedded computer vision application, (2007), pp. 1–6
16. S Calderara, R Cucchiara, A Prati, in *Proc. the 5th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS)*. Action signature: A novel holistic representation for action recognition, (2008), pp. 121–128
17. H Fujiyoshi, AJ Lipton, T Kanade, Real-time human motion analysis by image skeletonization. IEICE Trans. Inform. Syst. **87**(1), 113–120 (2004)
18. H-S Chen, H-T Chen, Y-W Chen, S-Y Lee, in *Proc. the 4th ACM International Workshop on Video Surveillance and Sensor Networks*. Human action recognition using star skeleton, (2006), pp. 171–178
19. H Ning, W Xu, Y Gong, T Huang, in *Computer Vision–ECCV 2008*. Latent pose estimator for continuous action recognition, (2008), pp. 419–433
20. L Zhu, Y Chen, Y Lu, C Lin, A Yuille, in *Proc. the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*. Max margin and/or graph learning for parsing the human body, (2008), pp. 1–8
21. M Raptis, D Kirovski, H Hoppe, in *Proc. the ACM SIGGRAPH/Eurographics Symposium on Computer Animation*. Real-time classification of dance gestures from skeleton animation, (2011), pp. 147–156
22. J Aggarwal, L Xia, Human activity recognition from 3D data: a review. Pattern Recognit. Lett. **48**, 70–80 (2014)
23. K Biswas, SK Basu, in *Proc. the 5th IEEE International Conference on Automation, Robotics and Applications (ICARA)*. Gesture recognition using Microsoft Kinect, (2011), pp. 100–103
24. W Li, Z Zhang, Z Liu, in *Proc. the IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*. Action recognition based on a bag of 3D points, (2010), pp. 9–14
25. P Dollár, V Rabaud, G Cottrell, S Belongie, in *Proc. the 2nd Joint IEEE International Workshop on Visual Surveillance and Performance Evaluation of Tracking and Surveillance*. Behavior recognition via sparse spatio-temporal features, (2005), pp. 65–72
26. O Oreifej, Z Liu, in *Proc. the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*. HON4D: Histogram of oriented 4D normals for activity recognition from depth sequences, (2013), pp. 716–723
27. H Rahmani, A Mahmood, DQ Huynh, A Mian, in *Proc. the European Conference on Computer Vision (ECCV)*. HOPC: Histogram of oriented principal components of 3D pointclouds for action recognition, (2014), pp. 742–757
28. CMU CMU, CMU Graphics Lab Motion Capture Database (2015). http://mocap.cs.cmu.edu/. Accessed 1 June 2016
29. F Lv, R Nevatia, in *Computer Vision–ECCV*. Recognition and segmentation of 3-D human action using HMM and multi-class Adaboost, (2006), pp. 359–372
30. M Martínez-Zarzuela, F Díaz-Pernas, A Tejeros-de-Pablos, D González-Ortega, M Antón-Rodríguez, Action recognition system based on human body tracking with depth images. Adv. Comput. Sci. Int. J. **3**(1), 115–123 (2014)
31. GA Carpenter, S Grossberg, N Markuzon, JH Reynolds, DB Rosen, Fuzzy ARTMAP: a neural network architecture for incremental supervised learning of analog multidimensional maps. IEEE Trans. Neural Netw. **3**(5), 698–713 (1992)
32. JA Suykens, J Vandewalle, Least squares support vector machine classifiers. Neural Process. Lett. **9**(3), 293–300 (1999)
33. A Veeraraghavan, A Srivastava, AK Roy-Chowdhury, R Chellappa, Rate-invariant recognition of humans and their activities. IEEE Trans. Image Process. **18**(6), 1326–1339 (2009)
34. M Müller, T Röder, in *Proc. the ACM SIGGRAPH/Eurographics Symposium on Computer Animation*. Motion templates for automatic classification and retrieval of motion capture data, (2006), pp. 137–146
35. J Wang, Y Wu, in *Proc. the IEEE International Conference on Computer Vision (ICCV)*. Learning maximum margin temporal warping for action recognition, (2013), pp. 2688–2695
36. H Pazhoumand-Dar, C-P Lam, M Masek, Joint movement similarities for robust 3D action recognition using skeletal data. J. Visual Commun. Image Represent. **30**, 10–21 (2015)
37. B Amor, J Su, A Srivastava, Action recognition using rate-invariant analysis of skeletal shape trajectories. IEEE Trans. Pattern Anal. Mach. Intell. **38**(1), 1–13 (2016)
38. J Liu, L Zhong, J Wickramasuriya, V Vasudevan, uWave: Accelerometer-based personalized gesture recognition and its applications. Pervasive Mobile Comput. **5**(6), 657–675 (2009)
39. P Mistry, P Maes, L Chang, in *Proc. the CHI Extended Abstracts on Human Factors in Computing Systems*. WUW-wear Ur world: a wearable gestural interface, (2009), pp. 4111–4116
40. M Müller, A Baak, S Hans-Peter, in *Proc. the ACM SIGGRAPH/Eurographics Symposium on Computer Animation*. Efficient and robust annotation of motion capture data, (2009), pp. 17–26
41. Z Prekopcsák, P Halácsy, C Gáspár-Papanek, in *Proc. the 10th ACM International Conference on Human Computer Interaction with Mobile Devices and Services*. Design and development of an everyday hand gesture interface, (2008), pp. 479–480
42. Z Zhang, Microsoft Kinect sensor and its effect. MultiMedia, IEEE. **19**(2), 4–10 (2012)
43. JM Chaquet, EJ Carmona, A Fernández-Caballero, A survey of video datasets for human action and activity recognition. Comput. Vis. Image Understand. **117**(6), 633–659 (2013)
44. S Sempena, NU Maulidevi, PR Aryan, in *Proc. the IEEE International Conference on Electrical Engineering and Informatics (ICEEI)*. Human action recognition using dynamic time warping, (2011), pp. 1–5
45. G Al-Naymat, S Chawla, J Taheri, in *Proc. the 8th Australasian Data Mining Conference-Volume 101*. SparseDTW: a novel approach to speed up dynamic time warping, (2009), pp. 117–127
46. J Martens, I Sutskever, in *Proc. the 28th International Conference on Machine Learning (ICML)*. Learning recurrent neural networks with Hessian-free optimization, (2011), pp. 1033–1040